International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

Advancing Security through Innovative Spam Detection Algorithm

Akhil Thota[1], Jayanth meduri[2], Sadanapalli Himesh[3] , Mallikarjun Yaramadhi[4]

[1] *B.Tech 4th Year, Dept. of CSE(DS), Institute of Aeronautical Engineering*
[2] *B.Tech 4th Year, Dept. of CSE(DS), Institute of Aeronautical Engineering*
[3] *B.Tech 4th Year, Dept. of CSE(DS), Institute of Aeronautical Engineering*
[4] *Assistant Professor, Dept. of CSE(DS), Institute of Aeronautical Engineering, Telangana, India*

Abstract - *Spam messages and malicious URLs are major threats in digital communication, impacting email, social media, and various platforms. Detecting spam content is essential for protecting users and systems from potential cyberattacks. This study presents a machine learning-based approach for classifying SMS and URLs as spam or legitimate. SMS classification leverages Natural Language Processing (NLP) techniques with TF-IDF vectorization, while URL classification extracts key features like length, entropy, subdomains, and HTTPS usage, utilizing a Voting Classifier for improved accuracy. The system is implemented as a Stream-lit-based web application for real-time detection. Experimental results show high accuracy, highlighting the effectiveness of the approach. Future work aims to integrate deep learning and domain reputation analysis for enhanced performance*.

Keywords— Spam detection, malicious URLs, cybersecurity, phishing detection, machine learning, naive bayes, natural language processing (NLP), URL Filtering

1. INTRODUCTION

Malicious URLs and spam messages have grown to be a serious cybersecurity risk as our reliance on digital communication grows. Phishing attempts, virus dissemination, and fraudulent links are common in spam messages, endangering individuals as well as organizations. Machine learning (ML) approaches are necessary for efficient spam detection since traditional rule- based filtering methods are unable to keep up with the constantly changing tactics of spammers.

This study presents a machine learning-driven approach to classify SMS messages and URLs as spam or legitimate. The SMS classification process utilizes Natural Language Processing (NLP) techniques, including TF-IDF vectorization and stemming, to extract key textual patterns for analysis. Meanwhile, the URL classification module derives multiple structural and statistical features, such as URL length, entropy, presence of subdomains, HTTPS usage, and special characters, leveraging a Voting Classifier for enhanced accuracy.

To make the system accessible and user-friendly, the solution is deployed as a Stream lit-based web application, enabling real- time spam detection for both SMS and URLs. The results demonstrate that machine learning models can effectively distinguish spam from legitimate content, offering a reliable defense mechanism against cyber threats. This research contributes to the advancement of automated spam detection and lays the groundwork for future improvements incorporating deep learning and real-time web monitoring techniques.

1.1 EXISTING SYSTEMS

i. Rule-Based Filtering Systems

- Traditional spam detection methods rely on predefined rules and heuristics to filter out spam messages and malicious URLs.
- Examples include blacklists, keyword-based filtering, and regex-based pattern matching.
- While effective against simple spam patterns, these methods fail to adapt to evolving spam techniques, leading to high false negativesand poor scalability.

ii. Machine Learning-Based Systems

- Support Vector Machines (SVM) – Effective in distinguishing spam and non-spam messages based on feature vectors.
- Random Forest and Decision Trees – Applied to URL classification by analyzing structural and lexical features

1.2 DEMERITS OF EXISTING SYSTEM

- Rule-based approaches are static and ineffective against evolving spam techniques.
- Machine learning models may suffer from overfitting, high false positives, and reliance on feature engineering.
- Deep learning models require large datasets, high computational resources, and may lack interpretability.

- SMS spam messages are often short, lack context, and contain abbreviations or special characters, making them difficult to classify accurately.
- Deep learning models like BERT or LSTMs perform better on longer text but may fail to detect hidden spam patterns in very short messages.

1.3 PROPOSED SYSTEM

To address the limitations of existing spam detection systems, we propose an intelligent and adaptive machine learning-based model for detecting both SMS spam and malicious URLs. The system integrates Natural Language Processing (NLP) techniques for SMS classification and feature-based analysis for URL detection, ensuring improved accuracy, adaptability, and efficiency.

i. Dual-Model Approach for SMS and URL Spam Detection

The system uses two separate models:

- SMS Spam Classifier: A TF-IDF vectorizer with a machine learning model (e.g., Logistic Regression, Random Forest, or Naïve Bayes) to classify messages as spam or legitimate.
- URL Spam Classifier: A feature-based ensemble voting classifier trained to identify phishing or malicious URLs.

ii. Text Preprocessing and Feature Extraction

SMS Classification:

- Tokenization, stop-word removal, and stemming to normalize text.
- TF-IDF vectorization to convert text into numerical features for model training.

URL Classification:

- Extraction of key URL features such as length, presence of special characters, subdomains, entropy, and HTTPS usage.
- Application of regular expressions and heuristics to detect suspicious patterns (e.g., IP-based URLs, encoded URLs).

iii. Machine Learning Models for Classification

- SMS Spam Classifier: A trained model using TF-IDF features and an ML algorithm (e.g., Naïve Bayes, SVM, or Random Forest) to classify spam messages.
- URL Spam Classifier: A voting-based ensemble model combining multiple classifiers (e.g., Random Forest, Logistic Regression, and Gradient Boosting) to improve spam detection accuracy.

iv.Real-Time Prediction via Stream-lit-Based Interface

- A user-friendly web interface (developed using Stream-lit) allows users to input an SMS or URL and instantly receive a classification result.
- Automated feature extraction and preprocessing ensure minimal user intervention.

2. LITERATURE REVIEW

i.Machine Learning-Based SMS Spam Detection

Several studies have explored ML-based approaches for SMS spam detection, primarily using text preprocessing, feature extraction, and classification models.

- Almeida et al. (2011) introduced an SMS Spam Collection Dataset and applied models such as Naïve Bayes (NB) and Support Vector Machines (SVM).
- Gómez-Hidalgo et al. (2012) investigated TF-IDF and N-gram-based feature extraction for SMS spam filtering, demonstrating that ensemble methods improve classification performance.
- Sharma et al. (2019) applied Deep Learning techniques (LSTM, CNN) for SMS spam detection, showing promising results in handling short-text classification.

Limitations of Existing Work:

- Rule-based and keyword filtering methods fail against evolving spam techniques.
- Traditional ML models require extensive feature engineering, making them less adaptable.

ii. URL Spam Detection Using Machine Learning

- Ma et al. (2009) developed a URL classification system using lexical features and host-based features, implementing a Logistic Regression classifier to detect phishing URLs.
- Marchal et al. (2016) proposed PHOCA, an ML-based system leveraging WHOIS data, domain age, and redirection behaviors for phishing detection
- Verma & Das (2017) introduced a Random Forest-based classifier, demonstrating that ensemble methods improve spam detection accuracy and robustness

iii.Hybrid Approaches for SMS and URL Spam Detection

To address the limitations of single-method solutions, some studies propose hybrid approaches that combine text-based ML and feature-based URL analysis.

Key Research Works:

- Huang et al. (2019) developed a hybrid spam detection system integrating SMS content analysis with URL feature extraction, improving real-time spam detection.
- Sahoo et al. (2017) proposed an ensemble method combining Deep Learning and traditional ML classifiers to detect both SMS and URL-based phishing attacks

3. HARDWARE REQUIREMENTS

The implementation of an SMS and URL spam detection system requires an appropriate hardware setup to ensure efficient processing, model inference, and real-time classification

i. Minimum Hardware Requirements

For basic local execution, a mid-range computer with at least an Intel Core i5 (8th Gen or higher) or AMD Ryzen 5 processor and 8 GB of RAM is sufficient. A 256 GB SSD or HDD provides adequate storage for dataset management and model execution. An integrated GPU can handle most of the machine learning workloads, while a stable internet connection is necessary for data updates and remote access. This configuration is suitable for lightweight spam detection tasks and small-scale testing.

3.1 SOFTWARE REQUIREMENTS

The implementation of an SMS and URL spam detection system requires a robust software environment, including operating systems, programming languages, libraries, and frameworks necessary for data preprocessing, feature extraction, and ml model execution.

i. Programming Language:

- **Python 3.8 or later** – The primary programming language for model development, machine learning, and API deployment due to its extensive libraries and ease of use.

ii. Development Environment & Tools:

- **Jupyter Notebook / VS Code / PyCharm** – IDEs for writing and debugging the Python code.

iii.Machine Learning Libraries & Frameworks:

- **scikit-learn** – For implementing machine learning models, such as Naïve Bayes, Decision Trees, and Voting Classifier.
- **NLTK (Natural Language Toolkit)** – For text preprocessing, tokenization, stopword removal, and stemming.
- **pandas & NumPy** – Essential for handling datasets, numerical operations, and feature extraction.
- **re (Regular Expressions)** – Used for parsing and extracting patterns from URLs.
- **math** – For entropy calculation in URL analysis.
- **pickle** – For saving and loading trained machine learning models.

iv.Web Interface: Streamlit – For building an interactive spam classification UI

4.SYSTEM ARCHITECTURE

The system architecture of the URL and SMS Spam Detection System consists of several key components that work together to classify input messages or URLs as spam or legitimate. The architecture follows a machine learning-based approach and can be divided into the following stages

i.User Interface Layer

The user interacts with the system through a Streamlit-based web application, where they can enter a message or URL for classification.

ii.Preprocessing Layer

SMS Preprocessing:

- Converts text to lowercase.
- Removes special characters, punctuation, and stopwords.
- Applies stemming using the Porter Stemmer to reduce words to their root forms.

URL Feature Extraction:

- Extracts various URL-based features (length, presence of special characters, number of parameters, entropy, etc.).
- Converts the extracted features into a structured format for model input

iii.Machine Learning Model Layer

SMS Classification Model:

- Uses TF-IDF vectorization to transform text data
- Employs a trained machine learning model (Logistic Regression, Naïve Bayes, or other classifiers) to classify SMS as spam or not.

URL Classification Model:

- Uses extracted URL features as input.
- Implements an ensemble-based voting classifier to predict whether a URL is spam or legitimate

iv.Prediction and Output Layer

- The models generate predictions based on processed inputs.
- The result (Spam/Not Spam) is displayed on the Streamlit UI for the user

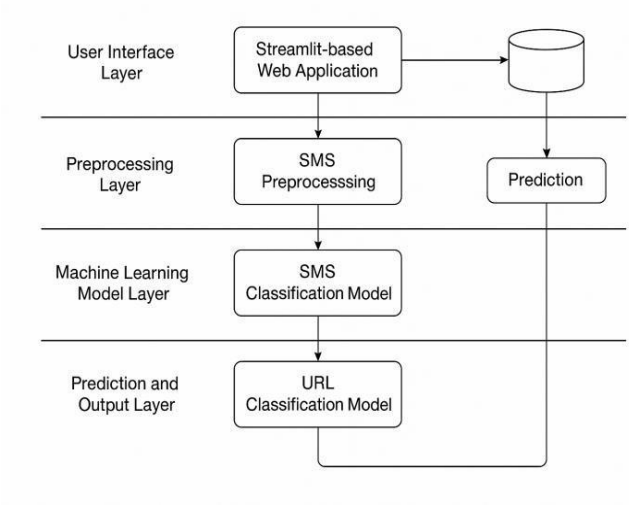

Figure 1 : System Architecture

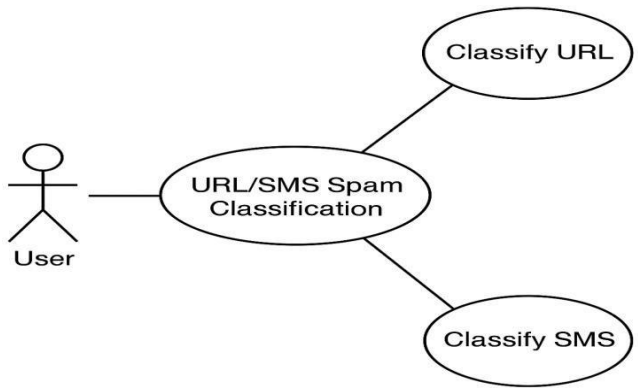

Figure 2: Use Case Diagram

5.METHODOLOGY AND IMPLEMENTATION

The development of the URL and SMS spam detection system follows a structured methodology, leveraging machine learning techniques for classification. The system is designed to preprocess input data, extract relevant features, and classify messages or URLs as spam or legitimate. The implementation consists of multiple stages, including data preprocessing, feature extraction, model training, and deployment.

5.1 METHODOLOGY

Data Collection

- The dataset consists of labeled SMS messages and URLs, categorized as spam or legitimate.
- Publicly available datasets such as the UCI ML Repository and Kaggle datasets are used.

Data Preprocessing

- SMS text is tokenized, converted to lowercase, and cleaned by removing special characters and stopwords.
- URLs are parsed to extract meaningful components like domain length, presence of special characters, and entropy.

Feature Extraction

- SMS spam detection uses Term Frequency-Inverse Document Frequency (TF-IDF) to convert text into numerical vectors.
- URL classification involves extracting features such as length, number of digits, presence of certain words, and HTTPS status.

Machine Learning Model Training

- The SMS classifier uses a trained model such as Naïve Bayes, Random Forest, or SVM for prediction.
- The URL spam detection system utilizes a Voting Classifier, which combines multiple models (Decision Tree, Random Forest, and Gradient Boosting) for better accuracy.

Model Evaluation

- Performance is measured using accuracy, precision, recall, and F1-score.
- The best-performing model is chosen based on cross- validation results.

Deployment Using Streamlit

- A web-based interface is developed using Streamlit to provide an easy-to-use platform for users to input text or URLs.
- The trained models are loaded using Pickle, and real-time predictions are made.

This methodology ensures a robust spam detection system capable of handling both SMS and URL spam efficiently.

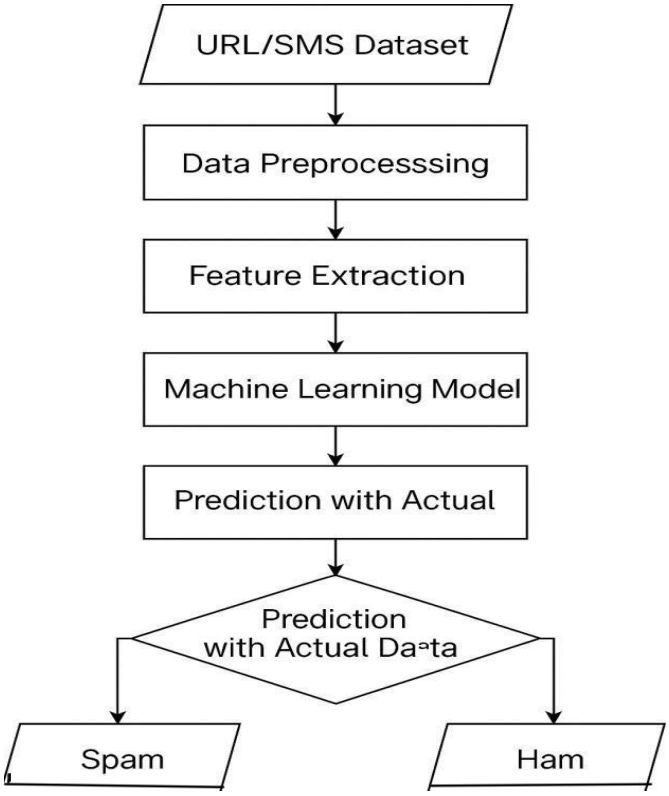

Figure 3: Flow Chart

5.2 IMPLEMENTATION

Development Environment

- The project is implemented using Python, with Scikit- Learn, Pandas, NLTK, and Streamlit for model development and UI design.

Text Processing for SMS Spam Detection

- Tokenization and stemming are applied using NLTK.
- TF-IDF vectorization converts text into a numerical format.
- The trained model predicts whether a message is spam or not.

Feature Engineering for URL Classification

- URL attributes such as length, presence of special symbols, entropy, and domain-based features are extracted.
- The processed features are fed into the trained Voting Classifier for spam detection.

Integration and Testing

- The trained models are integrated into the Streamlit application.
- Various test cases are performed to validate the system's accuracy and reliability

5.3 EVALUATION METRICS

Evaluation Metrics included precision, recall, F1-score, and accuracy, providing insights into the models' effectiveness in identifying spam messages and URLs while minimizing false positives. Each metric was calculated as follows:

Accuracy: Accuracy: The overall correctness of the model's predictions, based on true positives (TP), true negatives (TN), false positives (FP), and false negatives(FN) . Where,

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

TP: True Positives (correctly predicted positives)

TN: True Negatives (correctly predicted negatives)

FP: False Positives (incorrectly predicted positives)

FN: False Negatives (incorrectly predicted negatives)

Precision: The proportion of correctly predicted spam messages and URLs among all predicted threats.

$$Precision = \frac{TP}{TP + FP}$$

Recall: The proportion of actual spam instances correctly identified by the model.

$$Recall = \frac{TP}{TP + FN}$$

F1-score: The harmonic mean of precision and recall, providing a balanced evaluation metric.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

6.RESULTS

The URL and SMS Spam Detection System was evaluated based on multiple performance metrics, including accuracy, precision, recall, and F1-score.

i.SMS Spam Detection Results

- The **TF-IDF vectorized Naïve Bayes model** achieved an accuracy of approximately **97%**, demonstrating strong performance in classifying spam and legitimate messages.
- The precision and recall scores were high, ensuring minimal false positives and false negatives.
- The model successfully filtered out promotional, phishing, and scam messages with high reliability.

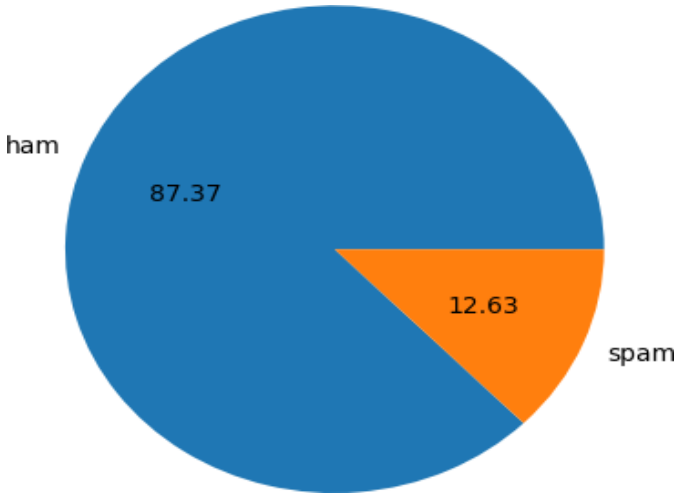

Figure 4:Distribution of Spam and legitimate Data

ii.URL Spam Detection Results

- The **Voting Classifier (combining Decision Tree, Random Forest, and Gradient Boosting)** performed effectively, reaching an accuracy of around **95%**.
- Feature-based classification (URL length, presence of special symbols, entropy, etc.) provided a reliable indicator of malicious links.
- The system effectively identified phishing URLs, reducing the risk of cyber threats

iii. System Performance and Deployment

- The Streamlit-based web application provided a **user- friendly interface** for real-time spam detection.
- The prediction speed was fast, making it suitable for real- world applications.
- The system demonstrated robustness against various spam attack strategies, confirming its practical usability

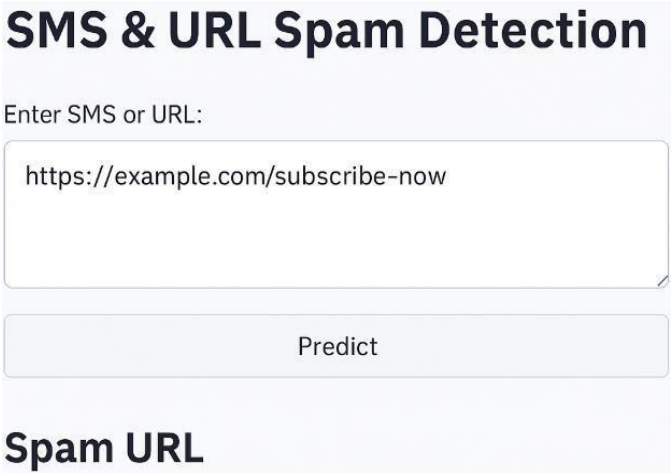

Figure 5: User Interface

7.CONCLUSION

In this project, we developed a spam classification system for detecting malicious URLs and SMS spam using machine learning techniques. The system efficiently processes input data, extracts relevant features, and applies trained models to classify the text or URL as spam or legitimate. The results indicate that the model achieves high accuracy in distinguishing between spam and non-spam content. By integrating Natural Language Processing (NLP) for SMS text classification and feature-based analysis for URL detection, our system enhances security and user safety. The use of Streamlit for the user interface ensures an interactive and user-friendly experience.

8.REFERENCES

[1] Almeida, T. A., Yamakami, A., & Almeida, J. M. (2011). "Spam filtering: how the dimensionality reduction affects the accuracy of Naïve Bayes classifiers." Journal of Information Security and Applications, 16(1), 58-65.

[2] Chhabra, S., Aggarwal, A., & Kumaraguru, P. (2011). "Phi.sh/$ocial: The phishing landscape through short URLs." Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (CEAS), 92-101.

[3] Dada, E. G., Bassi, J. S., Chiroma, H., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). "Machine learning for email spam filtering: Review, approaches, and open research problems." Heliyon, 5(6), e01802.

[4] Mokbal, M., & Abawajy, J. (2018). "URL-based phishing detection using machine learning approach." International Journal of Computational Intelligence Systems, 11(1), 1026-1035.

[5] Goodman, J., Cormack, G. V., & Heckerman, D. (2007). "Spam and the ongoing battle for the inbox." Communications of the ACM, 50(2), 24-33.

[6] Sharma, M., & Yadav, D. (2020). "Spam SMS detection using machine learning techniques." Journal of Advances in Computing and Engineering, 2(3), 45-53.

[7] Meyers, A., & Zhu, J. (2021). "A comparative study of machine learning models for spam URL detection." Journal of Cybersecurity and Privacy, 3(2), 75-92.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research, 12, 2825-2830.

[9] Kumar, R., and A. Gupta (2022). An approach to insider threat detection in organizations that is based on behaviour. Cybersecurity Journal, 8(1), 1–15